The DARPA Adaptive and Reflective Middleware Systems (ARMS) Program: Phase I

Pervasive Instrumentation and Adaptation for Distributed Real-Time Embedded Systems: Final Technical Report

Telcordia Technologies

Contract No.: NBCHC030132

April 4, 2005

Introduction

This is the final technical report on the Adaptive and Reflective Middleware Systems (ARMS) Phase I work by the Telcordia team. The Telcordia Team consists of Telcordia Technologies (Prime) and Prism Technologies (PrismTech), and the work was performed from October 1, 2003 to March 31, 2005. The report discusses architecture, implementation and validation aspects of the technology developed during Phase I. The report also includes a discussion of key results and lessons learnt.

Telcordia's focus in the ARMS program is in the development of an adaptive and reflective network Quality of Service (QoS) infrastructure for the Total Ship Computing Environment (TSCE) of next generation surface ships. Adaptive and reflective network QoS technology can play the vital role of providing ongoing, end-to-end assurance that mission-critical traffic has bounded queuing loss, delay, and jitter in the presence of changing load and network topology. The solution we developed is intended to work in an integrated fashion with resource managers proposed by other ARMS Technology Developers using the CORBA middleware and component technology and using the Multi-Layer Resource Management (MLRM) architecture framework. The MLRM framework was a major output from Phase I of the ARMS program and resulted from the combined efforts of the program participants. The purpose of the MLRM architecture is to push middleware technologies beyond current commercial capabilities. The current capabilities are largely limited to fixed static allocation of resources in support of predefined mission capabilities. Static allocations limit the ability of a military application to adapt to conditions that vary from the original system design. It is desirable for resource allocation to be performed dynamically and modified in response to faults, to changes in mission requirements, or to workload distributions that do not match the original mission-planning model.

Our solution is intended to support a growing network architecture trend to carry a mixture of traffic of varying importance, varying bandwidth, and varying delay sensitivity on a single IP network built from layer 2 and layer 3 technology, as illustrated in Figure 1.

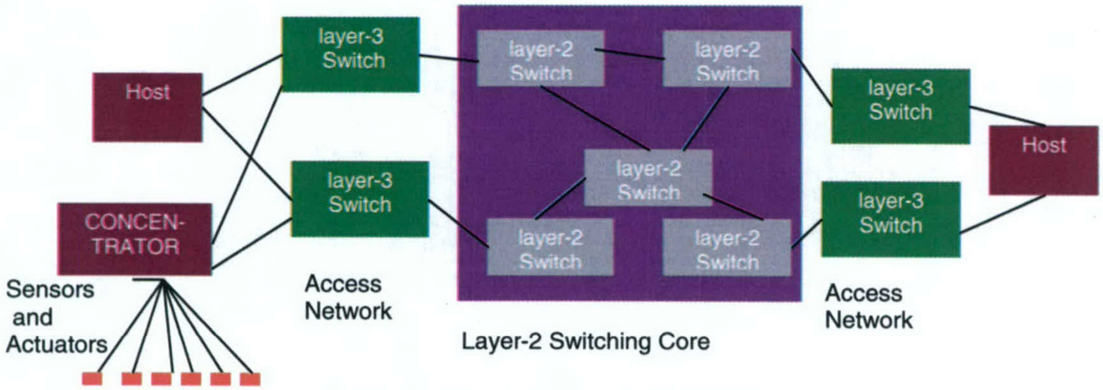

Figure 1: Illustration of Network Architecture

However, when such networks have moderate or full traffic loads, traditional best-effort techniques cannot assure acceptable performance and guarantees. Our adaptive and reflective QoS technology uses a Bandwidth Broker to provide admission control and enforcement using the Differentiated Services (DiffServ) and Class of Service (CoS) functionality of high-end COTS routers (at layer 3) and switches (at layer 2). The Bandwidth Broker technology adapts to changes in mission requirements, work load, and configuration by using discovery algorithms to maintain a current view of resource availability and traffic probes to detect the changing needs of high-priority flows. The Bandwidth Broker can assure good QoS for important flows, even in a fully loaded network.

Network QoS Architecture and Approach

Figure 2 illustrates our overall network QoS architecture. In Figure 2, R/S, ASM, IA and PM stand for Router/Switch, Application String Manager, Infrastructure Allocator and Pool Manager, respectively. ASM, IA and PM are higher-level MLRM components that are users of the network quality of service functionality provided by the Bandwidth Broker. The major logical components of the QoS management architecture are:

- Bandwidth Broker
- Flow Provisioner
- Performance Monitor
- Fault Monitor

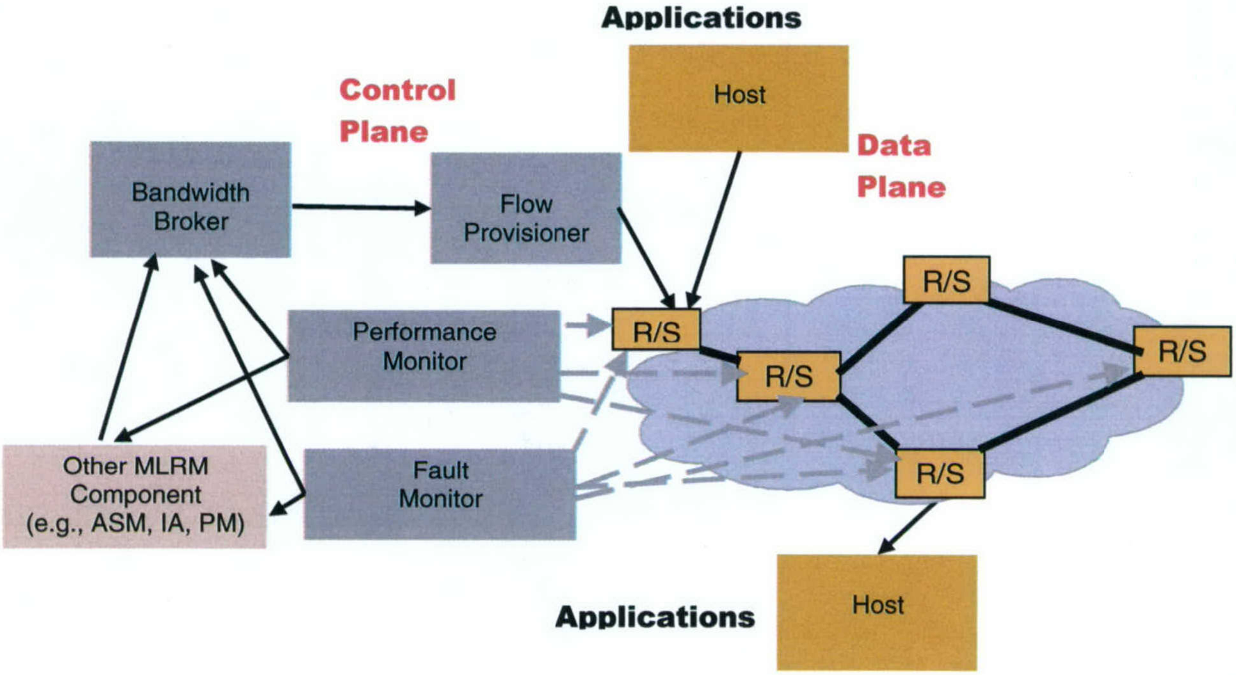

Figure 2: Network QoS Architecture

In Phase I of the program, we developed the Bandwidth Broker and Flow Provisioner. What was implemented in a limited way or not at all in our Phase I effort are two feedback mechanisms, fault monitoring (not implemented) and performance monitoring (partially implemented). In Phase I, the performance monitoring mechanism was limited to overload detection of mission-critical traffic. Our proposed Phase II work will focus on developing these two feedback mechanisms more fully. The fault monitor will both detect and react to faults, and the performance monitor will instrument for latency, jitter and available bandwidth metrics between any pair of endpoints in the network.

Bandwidth Broker

The basic functions provided by the Bandwidth Broker to higher-level MLRM components for the purpose of allocation and scheduling of mission tasks spanning the network are:

- Flow Admission Functions: Reserve, commit, modify, and delete flows (in support of distributed scheduling); and
- Queries: Bandwidth availability in different classes among pairs of pools and subnets (in support of allocation of processes to processors).

Bandwidth Broker: The Bandwidth Broker solution leverages DiffServ in layer-3 and CoS mechanisms in layer-2 network elements, in a transparent manner, to provide end-to-end QoS guarantees in a hybrid, heterogeneous environment. CoS mechanisms provide

functionality at layer 2 similar to what DiffServ mechanisms provide at layer 3[1]. They both provide aggregated traffic treatment in the core of the network and per-flow treatment at the edge of the network. Typical network implementations of QoS using DiffServ/CoS consist of the following steps:

1. At the ingress of the network, traffic is classified and marked as belonging to a particular class and may be policed or shaped to ensure that it does not exceed a certain rate or deviate from a certain profile.

2. In the network core, traffic is placed into different classes based on the marking of individual packets. Each class is provided treatment differentiated from all other classes but consistent for all packets within the class. This includes scheduling mechanisms that assign weights or priorities to different traffic classes (such as weighted fair queuing or priority queuing, respectively), and buffer management techniques that include assigning relative buffer sizes for different classes and packet-discard algorithms such as Random Early Detection (RED) and Weighed Random Early Detection (WRED).

Another popular mechanism to realize network QoS that is not used by the Bandwidth Broker is Integrated Services (IntServ). In IntServ, every router makes the decision whether or not to admit a flow with a given QoS requirement. Each router keeps the status of all admitted flows as well as the remaining available (uncommitted) bandwidth on its links. Some drawbacks with conventional implementations of IntServ are that (1) it does not scale well; (2) it does not lend itself well to centralized, high level policy-based management; and (3) it is applicable only to layer-3 IP networks. Our network QoS does not have any of these drawbacks.

DiffServ/CoS features by themselves are insufficient to guarantee end-to-end network QoS, because the traffic presented to the network must be made to match the network capacity. The main function of the Bandwidth Broker then is adaptive and reflective admission control that ensures there are adequate network resources to match the needs of admitted flows. To do its job, the admission control entity needs to be aware of the path being traversed by each flow, track how much bandwidth is being committed on each link for each traffic class, and estimate whether the traffic demands of new flows can be accommodated. As such, path discovery in combined layer-2 and layer-3 network was a major area of focus in Phase I.

The path discovery process determines the physical links that an admitted flow will traverse. Our path discovery, where possible, calculates the exact path traversed by each admitted flow. In cases where exact calculations are difficult or impossible, our methods

---

[1] Layer-2 CoS support is somewhat restrictive in its support for QoS. Typically, layer 2 supports a 3-bit Class of Service (CoS) marking or eight classes as opposed 6-bit or 64 different classes in DiffServ. Moreover, CoS has limited support mechanisms for scheduling and buffer management. The DiffServ and CoS features are typically implemented in software and in ASIC (Application Specific Integrated Circuits) hardware, respectively.

are conservative in the sense that they overestimate the extent to which admitted flows use bandwidth on physical links. For instance, when equal-cost routes are present between a source-destination pair of nodes, a conservative algorithm that accounts for application flows (between the pair of needs) along every possible equal cost path is being employed. The layer 3 portions of paths is discovered using *traceroute* from end to end for the flow. When a layer-2 network is multiply connected, switches use a spanning tree algorithm to remove possible network loops by disabling selected links. We discover using SNMP MIBs which ports are being blocked by each layer-2 switch. (See [1] for more details.)

In our approach, the Bandwidth Broker is also responsible for overall coordination. For instance, the Bandwidth Broker is responsible for assigning the appropriate traffic class to each flow, and coordinating provisioning of complex parameters for policing, marking, scheduling, and buffer management, such that contracted flows obtain the promised end-to-end QoS.

Support for Delay Bounds: In Phase I, the Bandwidth Broker admission decision for a flow was based on whether or not there was enough bandwidth on each link traversed by the flow. Toward the end of Phase I, we developed the computational techniques to provide both deterministic and statistical delay bound guarantees [2]. These guarantees are based on relatively expensive computations of occupancy or utilization bounds for various classes of traffic, performed only at the time of network configuration/reconfiguration, and relatively inexpensive checking for a violation of these bounds at the time of admission of a new flow. Delay guarantees raise the level of abstraction of the Bandwidth Broker to the higher layer MLRM components and enable these components to provide better end-to-end mission guarantees.

Flow Provisioner

The Flow Provisioner translates technology-independent configuration directives generated by the Bandwidth Broker into vendor-specific router and switch commands to classify, mark and police packets belonging to a flow. To demonstrate the applicability of our network QoS technology in different lab or network environments, in Phase I, we implemented the Flow Provisioner for layer-3 IOS CISCO (e.g., CISCO 3600 routers), layer-2/3 Catalyst switches (e.g., CISCO 6500 switches), layer-2/3 IOS switches (e.g., CISCO 4507 switches) and Linux routers.

In the Phase II ARMS program, we plan to build upon these two basic network QoS components, Bandwidth Broker and Flow Provisioner, we have developed to be more reflective and adaptive. Our proposed work will focus on:

- fault monitoring to provide continued assurance of network QoS for mission-critical tasks in the presence of single-mode faults; and
- performance monitoring improve the timely adaptation to network performance with probes and instrumentation for delay, jitter, and available bandwidth; and

Increasing the flexibility of our guarantees by incorporating deadline support in flow admission decisions based on sound mathematical calculations will also be an area of focus in Phase II. Moreover, our Phase II proposed network QoS solution will also

address mode and security policy changes, specifically those affecting network QoS globally. A mode here is taken to mean a major operational situation such as normal, alert, and battle mode. Detecting and reacting to all these types of changes is the cornerstone of a truly adaptive, reflective network QoS solution.

Implementation and Design Details

The Bandwidth Broker and Flow Provisioner make use of several open source technologies. They are implemented in Java and use the JacORB Object Request Broker, and the OpenCCM CORBA Component Model (CCM). The relational DBMS mySQL is the persistence mechanism used to recover from process and processor failures.

To support an arbitrary network deployment configuration, a network topology is input to the Bandwidth Broker through an XML file. The JAXB tools, another major free source software in the Java and XML space, are used to parse the input and build the required Java objects for switches, their interconnections (interfaces), and router-subnet relations.

The core of the Bandwidth Broker was designed and developed using UML. A UML model was developed to represent the network inventory and its state as required by the Bandwidth Broker. We also developed a method to translate a UML model into database tables and the Java code to access those tables. The method uses design patterns that allow one to write minimal SQL code. The SQL code exists only in base classes. As such, the code remains easily extensible to accommodate any changes to the UML model, such as addition of tables, addition of attributes and addition of relationships, and changes in the persistence mechanism employed.

Experimentation and Validation

The Bandwidth Broker and Flow Provisioner were involved in demonstrating their applicability for dynamic resource management to increase mission capacity. This is a key gate metric or challenge problem in Phase I. The basic steps in this gate metric experimentation were:

- Reserve bandwidth for certain key mission flows;
- Detect network overload in these key mission tasks (say in view of increased threats); and
- Re-reserve for increased bandwidth.

We implemented network overload detection, and overload "event" generation capabilities using the CCM framework. Since the Bandwidth Broker's admission capability would not allow overload in the network, excess offered loads have to be detected at the ingress of the network. This was easily accomplished as a direct extension to our Flow Provisioner capability that sets up the policing attributes for various flows at the ingress of the network. The monitoring program uses the policing functions provided by an ingress network element to determine the rate of packets being dropped or marked down. From these statistics, our monitoring program can detect that the offered load for a flow exceeds its provisioned limits during several contiguous intervals (the number of contiguous intervals can be set as a configuration parameter) and raise an overload event.

Major Results

The main result of our Phase I work is that it will lead to adaptive and reflective network resource management integrated into an overall adaptive and reflective resource management system for the TSCE which will enable more effective use of a ship's computing resources in dynamically changing and possibly hostile circumstances. Compared to the pre-existing static approach, this offers the potential for more effective execution of the ship's mission. The results achieved so far, especially in comparison with the current state-of-the-art and alternative approaches, include:

- Unified Layer-3/Layer-2 QoS Treatment: Our solution provides an integrated QoS treatment for heterogeneous layer-2 and layer-3 networks that can be centrally directed and policy-driven, and is more scalable than another commonly used QoS technique. The two main technologies for providing differentiated treatment of traffic are DiffServ/CoS and Integated Services (IntServ). The Bandwidth Broker makes use of DiffServ/CoS. In IntServ, every router makes the decision whether or not to admit a flow with a given QoS requirement. Some drawbacks with conventional implementations of IntServ are that (1) it requires per-flow state at each router, which can limit its scalability; (2) it makes its admission decisions based on local information rather than some adaptive, network-wide policy; and (3) it is applicable only to layer-3 IP networks.

- Flexibility in Admission Control: Our delay bounds calculations are set in a more generalized framework than what is found in the literature. Our calculations support any number of priority classes and within a priority class any number of weighted fair queuing classes. They are applicable for both layer-2 and layer-3 networks and flexible. The calculations support both deterministic and statistical guarantees using this generalized framework. Deterministic guarantees are usually applicable to the highest priority tasks, and statistical guarantees are more broadly applicable. The Bandwidth Broker currently supports capacity or bandwidth-based admission control. When these deterministic and statistical delay bound calculations are incorporated into the admission control process, our Bandwidth Broker will be very flexible in the mix of guarantees it can provide.

- Network QoS in an End-to-End Resource Management and CORBA Framework: The Bandwidth Broker technology has been applied for network QoS for sometime. Integration into middleware and integration into an end-to-end resource management framework so as to increase the level of network QoS abstraction to applications, however, have not been of focus in the past.

Lessons Learnt

- The focus on the MLRM architecture development in the early stage of the project and documenting the interfaces between various MLRM components both in IDL and UML did improve the overall understanding of the entire ARMS project and its scope, and communication among the ARMS participants.
- Emulab, the network emulation facility operated by the University of Utah, is a viable environment for integration. Emulab considerably reduced the dependency on

the AIF (Application Integration Facility) of Raytheon, the main laboratory designated for integration and demonstration. Emulab, however, can be used to test only layer-3 network QoS functionality. As such, the network QoS functionality testing may need to be augmented in laboratories other than AIF and Emulab in future phases.

- We believe that there was bit overemphasis on integration and the infrastructure for integration. This coupled with accelerated schedule in late summer of 2004 toward completing the gate metric demonstration in December 2004 did not leave ample time to be bit more innovative with resource management software and algorithms. A more balanced emphasis on integration and resource management functionality in various MLRM components should address this problem adequately in future phases.

References

[1] B. Dasarathy, S. Gadgil, R. Vaidyanathan, K. Parmeswaran, B. Coan, M. Conarty and V. Bhanot, Network QoS Assurance in a Multi-Layer Adaptive Resource Management Scheme for Mission-Critical Applications using the CORBA Middleware Framework, *Proc. RTAS 2005 (11th IEEE Real-Time and Embedded Technology and Application Symposium)*, pp. 246-255, March 7-10, 2005, San Francisco, CA.

[2] A. Neidhardt, B. Coan, and B. Dasarathy, Calculations for Admission Control, Telcordia ARMS Phase I Report, January 12, 2005 (under preparation for publication).

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*
04-04-2005

2. REPORT TYPE
Final Scientific Technical Report

3. DATES COVERED *(From - To)*
October 1, 2003 to March 31, 2005

4. TITLE AND SUBTITLE
The DARPA Adaptive and Reflective Middleware Systems (ARMS) Program: Phase I

Pervasive Instrumentation and Adaptation for Distributed Real-Time Embedded Systems: Final Technical Report

5a. CONTRACT NUMBER
NBCHC030132

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER
3GAR17 (Telcordia Internal Project Number)

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)
Balakrishnan Dasarathy and Brian Coan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Telcordia Technologies, Inc.
One Telcordia Drive
Piscataway, NJ 08854-4157

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
The Defense Advanced Research Projects Agency, Information Exploitation Office, Attn: Dr. Joseph Cross, Program Manager, Room 635, 3701 North Fairfax Dr., Arlington, VA 22203-1714

10. SPONSOR/MONITOR'S ACRONYM(S)
DARPA IXO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This is the final technical report on the Adaptive and Reflective Middleware Systems (ARMS) Phase I work by a team lead by Telcordia. Our focus in the ARMS program is in the development of an adaptive and reflective network Quality of Service (QoS) infrastructure for the Total Ship Computing Environment (TSCE) of next generation surface ships. Our technology uses a Bandwidth Broker to provide admission control, and leverages Differentiated Services and Class of Service functionality of high-end routers and switches for enforcement. The technology employs an integrated QoS treatment across a hybrid layer-2/layer-3 network and adapts to changes in mission requirements, work load, and configurations. This report discusses architecture, implementation, and validation aspects of the network QoS technology developed and lessons learned.

15. SUBJECT TERMS
adaptive reflective middleware, network QoS, admission control, bandwidth brokering, mission-critical systems

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**		9	Balakrishnan Dasarathy
					19b. TELEPHONE NUMBER *(Include area code)* 732-699-2430